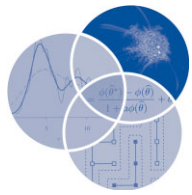
Focus Article



Steepest descent

Juan C. Meza*

The steepest descent method has a rich history and is one of the simplest and best known methods for minimizing a function. While the method is not commonly used in practice due to its slow convergence rate, understanding the convergence properties of this method can lead to a better understanding of many of the more sophisticated optimization methods. Here, we give a short introduction and discuss some of the advantages and disadvantages of this method. Some recent results on modified versions of the steepest descent method are also discussed. *WIREs Comp Stat* 2010 2 719–722 DOI: 10.1002/wics.117

INTRODUCTION

The classical steepest descent method is one of the oldest methods for the minimization of a general nonlinear function. The steepest descent method, also known as the gradient descent method, was first proposed by Cauchy.[1] In the original paper, Cauchy proposed the use of the gradient as a way of solving a nonlinear equation of the form

$$f(x_1, x_2, \ldots, x_n) = 0, \tag{1}$$

where f is a real-valued continuous function that never becomes negative and which remains continuous, at least within certain limits. The basis for the method is the simple observation that a continuous function should decrease, at least initially, if one takes a step along the direction of the negative gradient. The only difficulty then is deciding how to choose the length of the step one should take. While this is easy to compute for special cases such as a convex quadratic function, the general case usually requires the minimization of the function in question along the negative gradient direction.

Despite its simplicity, the steepest descent method has played an important role in the development of the theory of optimization. Unfortunately, the method is known to be quite slow in most real-world problems and is therefore not widely used. Instead, more powerful methods such as the conjugate gradient method or quasi-Newton methods are frequently used. Recently however, several attempts have been proposed to improve the efficiency of the method. These modifications have led to a newfound interest in the steepest descent method, both from a theoretical and practical viewpoint. These methods have pointed to the interesting observation that the gradient direction itself is not a bad choice, but rather that the original step length chosen leads to the slow convergence behavior.

*Correspondence to: jcmeza@lbl.gov

High Performance Computing Research, Lawrence Berkeley National Laboratory, 1 Cyclotron Road, Berkeley, CA, USA

DOI: 10.1002/wics.117

METHOD OF STEEPEST DESCENT

Suppose that we would like to find the minimum of a function $f(x), x \in R^n$, and $f : R^n \to R$. We will denote the gradient of f by $g_k = g(x_k) = \nabla f(x_k)$. The general idea behind most minimization methods is to compute a step along a given search direction, d_k, for example,

$$x_{k+1} = x_k + \alpha_k d_k, \quad k = 0, 1, \ldots, \tag{2}$$

where the step length, α_k, is chosen so that

$$\alpha_k = \arg\min_{\alpha} f(x_k + \alpha d_k). \tag{3}$$

Here arg min refers to the argument of the minimum for the given function. For the steepest descent method, the search direction is given by $d_k = -\nabla f(x_k)$. The steepest descent algorithm can now be written as follows:

Algorithm 1 Steepest Descent Method

Given an initial $x_0, d_0 = -g_0$, and a convergence tolerance *tol*
for $k = 0$ to *maxiter* **do**
  Set $\alpha_k = \text{argmin}\ \phi(\alpha) = f(x_k) - \alpha g_k$
  $x_{k+1} = x_k - \alpha_k g_k$
  Compute $g_{k+1} = \nabla f(x_{k+1})$
  if $||g_{k+1}||_2 \leq tol$ **then**
    converged
  end if
end for

The two main computational advantages of the steepest descent algorithm is the ease with which a computer algorithm can be implemented and the low storage requirements necessary, $O(n)$. The main work requirement is the line search required to compute the step length, α_k and the computation of the gradient.

CONVERGENCE THEORY

One of the main advantages to the steepest descent method is that it has a nice convergence theory.[2,3] It is fairly easy to show that the steepest descent method has a linear rate of convergence, which is not too surprising given the simplicity of the method. Unfortunately, even for mildly nonlinear problems this will result in convergence that is too slow for any practical application. On the other hand, the convergence theory for the steepest descent method is extremely useful in understanding the convergence behavior of more sophisticated methods.

To start, let us consider the case of minimizing the following quadratic function

$$f(x) = \frac{1}{2}x^T Q x - b^T x, \tag{4}$$

where $b \in R^n$, and Q is an $n \times n$ symmetric positive definite matrix. Since Q is symmetric and positive definite, all of the eigenvalues are real and positive. Let the eigenvalues of the matrix Q be given by $0 < \lambda_1 \leq \lambda_2 \cdots \leq \lambda_n$. Note that the gradient of (4) is simply

$$g(x) = Qx - b. \tag{5}$$

so we can write one step of the method of steepest descent as

$$x_{k+1} = x_k - \alpha_k(Qx_k - b), \tag{6}$$

where α_k is chosen to minimize $f(x)$ along the direction $-g_k$. A simple calculation (for the quadratic case) yields the following equation for α_k:

$$\alpha_k = \frac{g_k^T g_k}{g_k^T Q g_k}. \tag{7}$$

To analyze the convergence, it is easiest to consider the quantity $f(x_k) - f(x^*)$, where x^* denotes the global minimizer of Equation (4). Here we will follow proofs that can be found in standard texts such as[2,3]: We first notice that the unique minimizer to Equation (4) is given by the solution to the linear system

$$Qx^\star = b. \tag{8}$$

Consider the quantity:

$$\begin{aligned} f(x_k) - f(x^*) &= \frac{1}{2}\left(x_k^T Q x_k - b^T x_k\right) \\ &\quad -\frac{1}{2}\left((x^*)^T Q x^* - b^T x^*\right) \\ &= \frac{1}{2}\left(x_k^T Q x_k - (Qx^*)^T x_k\right) \\ &\quad -\frac{1}{2}\left((x^*)^T Q x^* - (Qx^*)^T x^*\right) \\ &= \frac{1}{2}(x_k - x^*)^T Q (x_k - x^*). \end{aligned}$$

To compute a bound, one uses a lemma due to Kantorivich, which can be found in Luenberger[2]. In particular, when the method of steepest descent with exact line searches is used on a strongly convex quadratic function then one can show that:

$$f(x_{k+1}) - f(x^*) \leq \left[\frac{\kappa(Q) - 1}{\kappa(Q) + 1}\right]^2 f(x_k) - f(x^*). \tag{9}$$

where $\kappa(Q) = \lambda_n/\lambda_1$ is the condition number of the matrix Q. A similar bound can be derived for the case of a general nonlinear objective function, if we assume that α_k is the global minimizer along the search direction.

EXAMPLE

Consider a simple example of a three-dimensional quadratic function given by

$$f(x) = \frac{1}{2}x^T Q - b^T x, \tag{10}$$

where

$$Q = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \tau & 0 \\ 0 & 0 & \tau^2 \end{pmatrix}, \quad b = -\begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}.$$

Using the steepest descent algorithm on this example problem produces the following results. The convergence tolerance was set so that the algorithm would terminate when $||g(x_k)|| \leq 10^{-6}$. One can clearly see the effects of even a mildly large condition number as predicted by the error bound and as seen in the number of iterations required to achieve convergence in Table 1.

TABLE 1 | Steepest Descent

τ	# iterations	$\kappa(A)$	Bound
2	27	4	.3600
5	161	25	.8521
10	633	100	.9801
20	2511	400	.9950
50	15,619	2500	.9984

$\kappa(A) = \lambda_1/\lambda_n$.

SCALING

One of the most important aspects in minimizing real-world problems is the issue of scaling. Because of the way that many scientific and engineering problems are initially formulated it is not uncommon to have difficulties due to variables having widely differing magnitudes. This can be due to many issues, but a common one is that variables have different physical units that can lead to the optimization variables having orders of magnitude differences. For example, one variable could be given in kilometers (10^3 m) and another variable might be in microseconds (10^{-6} s) leading to a difference of nine orders of magnitude. As a general rule of thumb, one would like to have all the variables in an optimization problem having roughly similar magnitudes. This leads to better search directions as well as in deciding when convergence is achieved. One fairly standard approach is to use a diagonal scaling based on what a "typical" value of a variable is expected to be. One would then transform the variables by the scaling:

$$\hat{x} = Dx, \tag{11}$$

where D is a diagonal scaling matrix. In the example given above, one simple choice would be:

$$D = \begin{pmatrix} 10^{-3} & 0 \\ 0 & 10^{6} \end{pmatrix}, \tag{12}$$

so that the components of $\hat{x}$ have similar magnitude.

EXTENSIONS

Recently, several new modifications to the steepest descent method have been proposed. In 1988, Barzilai and Borwein[4] proposed two new step sizes for use with the negative gradient direction. Although their method did not guarantee descent in the objective function values, their numerical results indicated a substantial improvement over the classical steepest descent method. One of their main observations was that the behavior of the steepest descent algorithm depended as much on the step size as on the search direction. They proposed instead the following procedure. First one writes the new iterate as:

$$x_{k+1} = x_k - \frac{1}{\alpha_k} g_k. \tag{13}$$

Then, instead of computing the step size by doing a line search or using the formula for the quadratic case 7, one computes the step length, α_k, through the following formula:

$$\alpha_k = \frac{s_{k-1}^T y_{k-1}}{s_{k-1}^T s_{k-1}}, \tag{14}$$

where $s_{k-1} = x_k - x_{k-1}$ and $y_{k-1} = g_k - g_{k-1}$. Using this new formula, Barzilai and Borwein were able to produce a substantial improvement in the performance of the steepest descent algorithm for certain test problems.

Subsequently, Raydan was able to prove convergence of the Barzilai and Borwein method for the case of a strictly convex quadratic function for any number of variables and in 1997 he proposed a nonmonotone line search strategy due to Grippo et al.'s[5] article, which guarantees global convergence[6] for the general nonlinear case. For an excellent overview on this subject and further details see Ref 7.

CONCLUSION

The steepest descent method is one of the oldest known methods for minimizing a general nonlinear function. The convergence theory for the method is widely used and is the basis for understanding many of the more sophisticated and well known algorithms. However, the basic method is well known to converge slowly for many problems and is rarely used in practice. Recent results have generated a renewed interest in the steepest descent method. The main observation is that the steepest descent direction can be used with a different step size than the classical method that can substantially improve the convergence. One disadvantage however is the lack of monotone convergence. After so many years, it is interesting to note that this method can still yield some surprising results.

REFERENCES

1. Cauchy A. Méthodes générales pour la résolution des systèmes d'équations simultanées. *C.R. Acad Sci Par* 1847, 25:536–538.
2. Luenberger DG, Ye Y. *Linear and Nonlinear Programming*. New York: Springer; 2008.
3. Nash SG, Sofer A. *Linear and Nonlinear Programming*. New York: McGraw-Hill; 1996.
4. Barzilai J, Borwein J. Two-point step size gradient methods. *IMA J Numer Anal* 1988, 8:141–148.
5. Grippo L, Lampariello F, Lucidi S. A nonmonotone line search technique for Newton's method. *SIAM J Numer Anal* 1986, 23:707–716.
6. Raydan M. The Barzilai and Borwein gradient method for the large scale unconstrained minimization problem. *SIAM J Optim* 1997, 7(1):26–33.
7. Fletcher R. On the Barzilai-Borwein method. In: Qi L, Teo K, Yang X, eds. *Optimization and Control with Applications*. New York: Springer; 2005, 235–256.

